LEVEL II

2 B.S.

ADA086101

TR-844 December, 1979
DAAG-53-76C-0138

FEATURE VALUE SMOOTHING AS AN AID IN TEXTURE ANALYSIS

Tsai-Hong Hong
Angela Y. Wu
Azriel Rosenfeld

Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

COMPUTER SCIENCE
TECHNICAL REPORT SERIES

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742

TR-844 December, 1979
DAAG-53-76C-0138

FEATURE VALUE SMOOTHING AS AN AID IN TEXTURE ANALYSIS

Tsai-Hong Hong
Angela Y. Wu
Azriel Rosenfeld

Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

ABSTRACT

When texture features are measured on small subimages, they are unreliable; but if we use large subimages, it is hard to find subimages that are uniformly textured. This paper describes a compromise approach: measure the features on small subimages, and smooth the resulting feature values in such a way that neighboring subimages that belong to differently textured regions are unlikely to influence one another. When this is done, classification performance improves substantially. Improvement is also obtained when the subimages are classified probabilistically and relaxation is used to adjust the class probabilities. The problem of choosing a window size that minimizes overall misclassification probability is also discussed.

The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (DARPA Order 3206) is gratefully acknowledged, as is the help of Kathryn Riley in preparing this paper.

1. Introduction

Given a set of image windows, each consisting of a single texture, the problem of texture classification is to classify the windows into texture types based on a set of feature measurements. A wide variety of statistical features can be used for texture classification; see [1] for a survey.

To obtain good classification performance, texture features should be computed for windows of adequate size, typically 64 by 64 pixels. If smaller windows are used, the feature values become unreliable, and classification performance deteriorates. However, the size of the windows required for reliable classification poses a problem when we try to analyze the textures on a given image; the larger the windows used, the fewer of them will fit inside the uniformly textured regions on the image, so that it becomes difficult to obtain sufficient numbers of uniformly textured samples.

This problem can be alleviated, to a substantial extent, by using small windows, and smoothing the texture feature values before using them for classification. As an alternative to iterative smoothing, one can use the initial feature values to probabilistically classify the windows and then use a relaxation scheme to adjust the class probabilities.

Sections 2-4 show how interactions between windows of a given size, in the form of iterative local smoothing or probabilistic relaxation, can be used to improve the reliability of texture features. Section 5 discusses the window sizes that should be used to minimize the number of misclassified pixels.

2. Data and features

2.1 Image data

In each experiment, a 512 by 512 pixel image was used, composed of two texture types with the 45° diagonal as the dividing line. Thus each image consists of 56 windows of size 64 by 64 that represent pure texture types (28 of each), plus 8 mixed windows lying on the diagonal. If we use windows of size 32 by 32 (divide each 64 by 64 window into four quadrants) we have 120 pure windows of each type and 16 mixed windows; while if we use size 16 by 16, we have 496 pure windows of each type and 32 mixed windows. Note that the mixed area is reduced as the windows get smaller; the number of pixels belonging to mixed windows is 2^{15} for 64 by 64's, 2^{14} for 32 by 32's, and 2^{13} for 16 by 16's.

In this study we used three LANDSAT geological terrain types, Mississippian limestone and shale (M), Pennsylvanian sandstone and shale (P), and Lower Pennsylvanian shale (L). Thus there were three 512 by 512 images, one containing each pair of terrain types (M and P, M and L, L and P). These images are shown in Figures 1-3.

Since texture feature values are computed for each individual window, we histogram flattened each 64 by 64 pure window, as well as the upper and lower pure triangles of each 64 by 64 mixed window on the diagonal, to remove any effects of unequal brightness and contrast. We did not flatten the 32 by 32 or 16 by 16 windows individually; the 32 by 32 flattened values would be quite similar to the 64 by 64 flattened values, and the 16 by 16 windows are somewhat too small for flattening.

2.2 Texture features

The texture feature used was the second-order gray level statistic "CONTRAST" [1], which is the moment of inertia of the cooccurrence matrix about its main diagonal; it was measured for a one-pixel displacement in the horizontal direction. The CONTRAST feature values were calculated for each of the 1024 16 by 16 windows. A 32 by 32 window is made up of four 16 by 16 quadrants. If we assume that border effects are minimal, then instead of calcluating the CONTRAST feature value from the co-occurrence matrix for each 32 by 32 window, we can obtain the feature value by averaging the values for the 16 by 16 windows in each of its four quadrants; and similarly, we can obtain it for a 64 by 64 window by averaging the four corresponding 32 by 32 window values. A comparison study showed that the discrepancy between these two methods was minimal, and the computationally cheaper method was therefore used.

The mean μ and standard deviation σ of the feature values for the windows of each size in each class are given in Table 1. These μ and σ values define a Gaussian probability density for each class and each size. We used these densities in conjunction with Bayes' theorem, to compute the probability that each window belongs to each of the two classes,

$$\mathrm{Prob}(C_i|x) = \frac{\mathrm{Prob}(x|C_i)}{\mathrm{Prob}(x|C_1) + \mathrm{Prob}(x|C_2)} \qquad i = 1,2,$$

and to classify the window according to the greater of these probabilities. Since the densities overlap, these maximum-likelihood classifications are not all correct. The error rates, defined to be the percentages of unmixed windows that were misclassified, are shown in the first row (iteration 0) of Table 2. The error rate is based on the unmixed windows since the mixed windows on the diagonal always have 50% error, no matter which way we classify them.

3. Smoothing technique

A smoothing process was iteratively applied to the feature values by selectively averaging the value measured for a given window with the values for some of the neighboring windows. For the windows along the image borders, it is assumed that there are extra rows and columns of windows outside the image whose feature values are the same as those of the windows on the border, as shown in Figure 4.

The averaging should be done in such a way that the neighbors used are likely to have the same texture as the given window. Image smoothing techniques such as median filtering (value replaced by the median of the feature values in the neighborhood) [2] and E^k schemes (average with the k neighbors whose feature values are closest to those of the given window) [3] with $k \leq 5$ have the desired behavior, namely, no problems arise for windows near the border between the two textures.

Under both the median filtering and E^5 smoothing process, the variability of the values within each class rapidly decreases; this results in substantially reduced error rates when the windows are classified on the basis of their smoothed values. Table 2 shows the error rates when applying the E^5 smoothing process to the images in Figures 1-3. We see that for the 64 by 64 windows, the error rate is reduced to almost zero immediately, while for the smaller windows, it is reduced to a level usually achievable only through the use of larger windows.

In the above results, the error rates were calculated on the basis of the mean and standard deviation of the smoothed values for each window size of each texture type, which defines updated Gaussian distributions for each iteration. Another possibility would be to use the Gaussian distributions derived from the original unsmoothed image. This would mean that the initial feature values are used as training samples and the smoothed values are test samples, while updating the Gaussians at each iteration means that the smoothed values are used both as the training and testing samples. Table 3 shows the error rates derived from non-updated Gaussians for the three images. Interestingly, the results are about as good as those obtained with updating, and the improvement is still very good, except for the 64x64 ML case, where the error rate first drops to zero and then increases again. It should be pointed out, however, that for such large windows, it is not meaningful to iterate the smoothing process more than a few times, since there are so few windows in the array.

Tables 4 and 5 show the error rates with and without updating the means and standard deviations when median filtering, rather than E^5, is used; the results are even better than those in Tables 2 and 3. For comparison purposes, Table 6 shows the error rates obtained when we smooth the feature values by averaging the value for each window with the value for all 8 neighboring windows (i.e., E^8). Problems arise for the windows near the diagonal border between the two texture types since such windows have five neighboring windows of the same texture type, but one of the other type and two of mixed types. This border effect is more significant for the larger windows since the percentage of these "problem windows" is higher, 25% for the 64 by 64 level, 12.5% for 32 by 32, and 6.25% for 16 by 16. This effect of this problem can be seen from the error rates in Table 6. One would expect the error rate for the 64 by 64 windows to be less than that for the 32 by 32's since the feature values are more reliable. Here, however, after 4 iterations, the error rate for the 64 by 64 windows was worse than that for the 32 by 32 windows.

4. Probabilistic relaxation

An alternative to iterative smoothing is to use the initial feature values to probabilistically classify the windows and then use a relaxation scheme to adjust the class probabilities for each window according to the class probabilities of its neighbors and the compatibilities between its class and its neighbors' classes. There are two probabilistic relaxation schemes, that initially introduced by Rosenfeld, Hummel, and Zucker [4], and that recently introduced by Peleg [5]. They used different class probability updating rules and different compatibility coefficient calculations. Tables 7 and 8 show that applying the relaxation schemes resulted in a reduced error rate. Note that the initial error rates are different from those in Tables 2-6 since no extra border windows were assumed and hence the windows at the image borders were not used in the calculation of the error rates.

The performance of the two relaxation schemes are comparable; the Rosenfeld, Hummel and Zucker method did slightly better for two of the images. For all three samples, the relaxation schemes do not work as well as smoothing, or at least do not converge as quickly. It should be mentioned that in analogous studies of pixel classification based on spectral signatures, relaxation gave better results than smoothing [6].

5. Window sizes

As pointed out in Section 1, texture feature values measured on small windows are not reliable, but using small windows has the advantage that more of them can fit inside a uniformly textured region. Moreover, the smaller windows can get closer to the border of the two texture types in the sense that fewer pixels are in the mixed windows, which automatically have half the pixels misclassified. The percentage of misclassified pixels is (half of the number of pixels in mixed windows + the number of pixels in the pure windows times window error rate) divided by the total number of pixels. Table 9 shows the percent misclassified pixels for each size window both initially and after up to 12 iterations for each sample based on the error rates in Table 3. Initially, 64 by 64 windows give fewer misclassified pixels; but after up to 12 iterations, the least number of misclassified pixels is given by 32x32 windows in two cases and by 16x16 windows in the other case.

6. Concluding remarks

This study shows that both iterative smoothing and relaxation processes improve the reliability of texture features measured over windows of different sizes. The higher rate of misclassified pure windows for smaller size windows is compensated by fewer pixels belonging to the mixed windows on the border of the two textures, so that in many instances, fewer pixels are misclassified when 32 by 32 windows are used instead of 64 by 64.

In this experiment, only features derived from windows of a single size were allowed to interact. Chen and Pavlidis [7] have described a split-and-merge method of texture segmentation in which feature values measured on windows of different sizes are compared. If the values for all four quadrants of a window are sufficiently close to the values for the entire window, then the window need not be subdivided; otherwise, we split it into quadrants. It would be desirable to combine their method of "vertical" interaction between the feature values with our "horizontal" interaction method.

References

1. R. M. Haralick, Statistical and structural approaches to texture, Proc. IEEE 67, 1979, 786-804.

2. W. K. Pratt, Median filtering, in Semiannual Report, Image Processing Institute, Univ. of Southern California, 1975, 116-123.

3. L. S. Davis and A. Rosenfeld, Noise cleaning by iterated local averaging, IEEETSMC-8, 1978, 705-710.

4. A. Rosenfeld, R. Hummel, and S. Zucker, Scene labeling by relaxation operations, IEEETSMC-6, 1976, 420-433.

5. S. Peleg, A new probabilistic relaxation scheme, IEEETPAMI-2, 1980, in press.

6. J. O. Eklundh, H. Yamamoto, and A. Rosenfeld, A relaxation method for multispectral pixel classification, IEEETPAMI-2, 1980, in press.

7. P. C. Chen and T. Pavlidis, Segmentation by texture using a co-occurrence matrix and a split and merge algorithm, CGIP 10, 1979, 172-182.

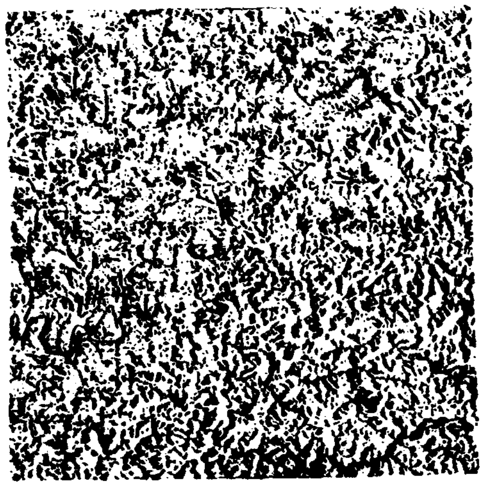

Figure 1. Geological terrain types from a LANDSAT frame. The upper left is Mississippian limestone and shale; the lower right is Pennsylvanian sandstone and shale: M/P.

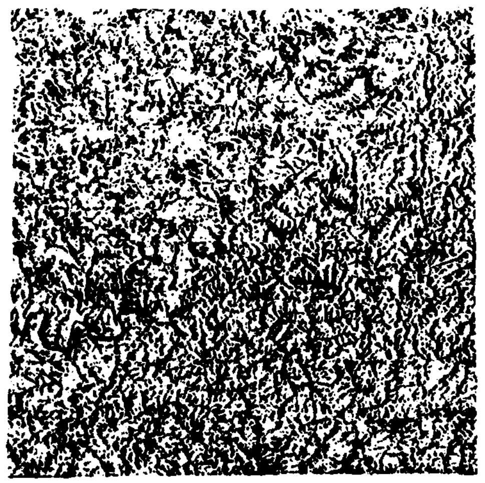

Figure 2. Same as Figure 1 except lower right is Lower Pennsylvanian shale: M/L.

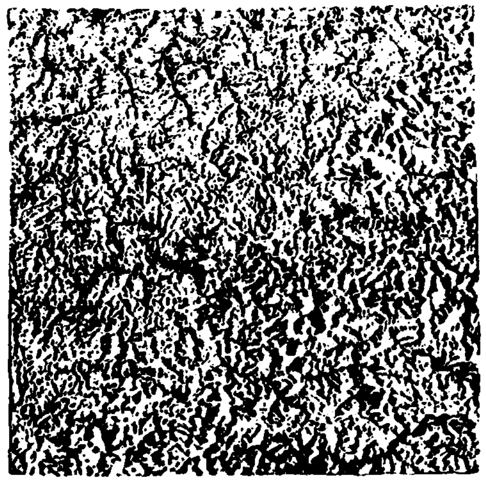

Figure 3. Same as Figure 1 except upper left is Lower Pennsylvanian shale: L/P.

$$\begin{array}{c|cccccc|c}
a_{22} & a_{21} & a_{22} & & a_{2,n-1} & a_{2n} & a_{2,n-1} \\
\hline
a_{12} & a_{11} & a_{12} & \cdots & a_{1,n-1} & a_{1n} & a_{1,n-1} \\
a_{22} & a_{21} & a_{22} & \cdots & a_{2,n-1} & a_{2n} & a_{2,n-1} \\
 & \vdots & \vdots & \ddots & & \vdots & \\
a_{n-1,2} & a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,n-1} & a_{n-1,n} & a_{n-1,n-1} \\
a_{n2} & a_{n1} & a_{n2} & \cdots & a_{n,n-1} & a_{nn} & a_{n,n-1} \\
\hline
a_{n-1,2} & a_{n-1,1} & a_{n-1,2} & & a_{n-1,n-1} & a_{n-1,n} & a_{n-1,n-1}
\end{array}$$

Figure 4. Feature values for neighbors of windows along the image borders.

Terrain Type		Window Size		
		64 by 64	32 by 32	16 by 16
Mississipian	μ	15.32	15.33	15.32
	σ	1.50	2.21	3.44
Pennsylvanian	μ	11.66	11.67	11.68
	σ	1.20	1.75	2.65
Lower Pennsylvanian	μ	18.53	18.53	18.46
	σ	1.14	2.19	3.64

Table 1. Means and standard deviations for the terrain types.

	M/P			M/L			L/P		
	Window Size			Window Size			Window Size		
Iteration	64x64	32x32	16x16	64x64	32x32	16x16	64x64	32x32	16x16
0	9	19	28	11	23	33	0	4	15
1	2	9	18	0	10	25	0	0	5
2	2	7	16	0	5	21	0	0	3
3	2	7	15	0	3	18	0	0	3
4	2	6	14	0	2	17	0	0	2
5	2	6	14	0	3	17	0	0	2
6	2	6	13	0	3	16	0	0	1
7	2	5	13	0	3	15	0	0	1
8	2	6	12	0	2	15	0	0	1
9	2	5	12	0	2	14	0	0	1
10	2	5	12	0	2	14	0	0	1
11	2	5	11	0	2	13	0	0	1
12	2	5	11	0	2	13	0	0	1

Table 2. Error rates (%) for the images in Figures 1-3 using the E^5 smoothing method.

	M/P			M/L			L/P		
	Window size			Window size			Window size		
Iteration	64x64	32x32	16x16	64x64	32x32	16x16	64x64	32x32	16x16
0	9	19	28	11	23	33	0	4	15
1	2	8	19	2	9	26	0	0	5
2	2	5	16	0	5	23	0	0	2
3	2	4	14	2	3	21	0	0	2
4	2	4	14	4	2	20	0	0	2
5	2	3	13	4	3	19	0	0	2
6	2	3	13	5	2	19	0	0	1
7	2	3	13	5	2	19	0	0	1
8	2	3	12	7	2	19	0	0	1
9	2	3	12	7	2	18	0	0	1
10	2	3	11	7	2	19	0	0	1
11	2	3	11	7	2	18	0	0	1
12	2	3	11	7	2	18	0	0	1

Table 3. Analogous to Table 2, but without updating the means and standard deviations at each iteration.

	M/P			M/L			L/P		
	Window Size			Window Size			Window Size		
Iteration	64x64	32x32	16x16	64x64	32x32	16x16	64x64	32x32	16x16
0	9	19	28	11	23	33	0	4	15
1	0	5	16	0	5	19	0	0	3
2	0	1	13	0	2	14	0	0	2
3	0	1	11	0	2	10	0	0	2
4	0	2	10	0	1	7	0	0	1
5	0	2	9	0	0	6	0	0	1
6	0	2	9	0	0	6	0	0	1
7	0	1	9	0	0	5	0	0	1
8	0	1	9	0	0	4	0	0	1
9	0	1	9	0	0	4	0	0	1
10	0	1	9	0	0	4	0	0	1
11	0	1	8	0	0	4	0	0	1
12	0	1	8	0	0	4	0	0	1

Table 4. Analogous to Table 2, using median filtering rather than E^5.

	M/P			M/L			L/P		
	Window Size			Window Size			Window Size		
Iteration	64x64	32x32	16x16	64x64	32x32	16x16	64x64	32x32	16x16
0	9	19	28	11	23	33	0	4	15
1	0	4	18	0	5	18	0	0	3
2	0	1	15	0	2	13	0	0	2
3	0	1	14	0	1	11	0	0	2
4	0	1	14	0	0	9	0	0	1
5	0	1	13	0	0	8	0	0	1
6	0	1	13	0	0	7	0	0	1
7	0	1	13	0	0	7	0	0	1
8	0	1	13	0	0	7	0	0	1
9	0	1	13	0	0	7	0	0	1
10	0	1	12	0	0	7	0	0	1
11	0	1	12	0	0	7	0	0	1
12	0	1	11	0	0	7	0	0	1

Table 5. Analogous to Table 3, using median filtering rather than E^5.

	Window Size		
Iteration	64x64	32x32	16x16
0	9	19	28
1	2	5	15
2	5	4	10
3	4	4	9
4	5	4	8
5	7	4	7
6	7	5	7
7	7	5	7
8	7	5	7
9	7	5	7
10	7	5	6
11	9	5	6
12	9	6	6

Table 6. Same as first part of Table 3 using averaging with all neighbors.

	M/P			M/L			L/P		
	Window Size			Window Size			Window Size		
Iteration	64x64	32x32	16x16	64x64	32x32	16x16	64x64	32x32	16x16
0	7	19	28	13	25	35	0	4	15
1	3	15	26	3	22	33	0	1	10
2	0	11	25	0	20	33	0	1	6
3	0	9	23	0	18	32	0	1	4
4	0	7	21	0	18	32	0	1	3
5	0	5	20	0	17	31	0	1	3
6	0	4	19	0	16	31	0	1	2
7	0	3	17	0	16	30	0	1	2
8	0	3	16	0	13	30	0	1	2
9	0	3	15	0	13	30	0	1	2
10	0	3	14	0	12	30	0	1	1
11	0	3	14	0	12	30	0	1	1
12	0	3	13	0	12	29	0	1	1

Table 7. Error rates (%) for the images in Figures 1-3 using the Rosenfeld, Hummel and Zucker relaxation scheme.

	M/P			M/L			L/P		
	Window Size			Window Size			Window Size		
Iteration	64x64	32x32	16x16	64x64	32x32	16x16	64x64	32x32	16x16
0	7	19	28	13	25	35	0	4	15
1	7	17	27	0	23	34	0	2	12
2	0	13	26	7	20	33	0	1	9
3	0	11	25	0	20	33	0	1	7
4	0	9	23	0	18	32	0	1	5
5	0	8	22	0	18	32	0	0	3
6	0	7	21	0	17	32	0	0	2
7	0	4	20	0	17	31	0	0	2
8	0	4	19	0	16	31	0	0	2
9	0	3	18	0	16	31	0	0	2
10	0	3	17	0	15	30	0	0	1
11	0	3	16	0	13	30	0	0	1
12	0	3	15	0	13	30	0	0	1

Table 8. Same as Table 7 except using the Peleg relaxation scheme.

	M/P			M/L			L/P		
	Window Size			Window Size			Window Size		
	64x64	32x32	16x16	64x64	32x32	16x16	64x64	32x32	16x16
Before Smoothing	14	21	29	16	25	34	6	7	16
After Smoothing	8	5	12	6	5	19	6	6	2

Table 9. Pixel Error Rates (%)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A086 101

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

(6) FEATURE VALUE SMOOTHING AS AN AID IN TEXTURE ANALYSIS.

5. TYPE OF REPORT & PERIOD COVERED

(9) Technical rept.,

6. ~~PERFORMING~~ ORG. REPORT NUMBER

(14) TR-844

7. ~~AUTHOR(s)~~

(10) Tsai-Hong Hong
Angela Y. Wu
Azriel Rosenfeld

8. CONTRACT OR GRANT NUMBER(s)

DAAG-53-76C-0138

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD 20740

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

(12) 23

11. CONTROLLING OFFICE NAME AND ADDRESS

U. S. Army Night Vision Laboratory
Fort Belvoir, VA 22060

12. ~~REPORT DATE~~

(11) Dece~~mber,~~ 1979

13. NUMBER OF PAGES

20

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

(15) DAAG53-76-C-Ø138,
DARPA Order-32Ø6

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Image processing
Pattern recognition
Texture analysis
Smoothing
Relaxation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

When texture features are measured on small subimages, they are unreliable; but if we use large subimages, it is hard to find subimages that are uniformly textured. This paper describes a compromise approach: measure the features on small subimages, and smooth the resulting feature values in such a way that neighboring subimages that belong to differently textured regions are unlikely to influence one another. When this is done, classification performance improves substantially. Improvement is also obtained when the subimages

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

411074

are classified probabilistically and relaxation is used to adjust the class probabilities. The problem of choosing a window size that minimizes overall misclassification probability is also discussed.